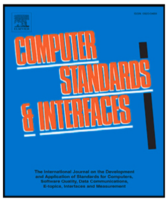
Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi



Energy consumption assessment in embedded AI: Metrological improvements of benchmarks for edge devices

Andrea Apicella [b], Pasquale Arpaia [a],*, Luigi Capobianco [d], Francesco Caputo [a], Antonella Cioffi [d], Antonio Esposito [a], Francesco Isgrò [a], Rosanna Manzo [c], Nicola Moccaldi [a], Danilo Pau [e], Ettore Toscano [d]

[a] Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Università degli Studi di Napoli Federico II, Naples, Italy
[b] Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica applicata (DIEM), Università degli Studi di Salerno, Fisciano, Italy
[c] Dipartimento di Sanità Pubblica e Medicina Preventiva, Università degli Studi di Napoli Federico II, Naples, Italy
[d] Software Design Center, STMicroelectronics, Marcianise, Italy
[e] System Research and Applications, STMicroelectronics, Agrate Brianza, Italy

ARTICLE INFO

Keywords:
Energy assessment
Embedded AI
Tiny-ML
Uncertainty analysis
Edge device benchmark

ABSTRACT

This manuscript proposes a new method to improve the MLCommons protocol for measuring power consumption on Microcontroller Units (MCUs) when running edge Artificial Intelligence (AI). In particular, the proposed approach (i) selectively measures the power consumption attributable to the inferences (namely, the predictions performed by Artificial Neural Networks — ANN), preventing the impact of other operations, (ii) accurately identifies the time window for acquiring the sample of the current thanks to the simultaneous measurement of power consumption and inference duration, and (iii) precisely synchronize the measurement windows and the inferences. The method is validated on three use cases: (i) Rockchip RV1106, a neural MCU that implements ANN via hardware neural processing unit through a dedicated accelerator, (ii) STM32 H7, and (iii) STM32 U5, high-performance and ultra-low-power general-purpose microcontroller, respectively. The proposed method returns higher power consumption for the two devices with respect to the MLCommons approach. This result is compatible with an improvement of selectivity and accuracy. Furthermore, the method reduces measurement uncertainty on the Rockchip RV1106 and STM32 boards by factors of 6 and 12, respectively.

1. Introduction

The rapid expansion of Internet of Things (IoT) devices has ushered in a new era of connected intelligence at the edge, where data processing, low latency, and real-time decision making can take place directly at the edge [1]. These IoT devices cover a variety of applications, from smart home sensors [2], to industrial automation [3], and health monitoring systems [4], where low latency responses and energy efficiency are essential.

Extending computation to more peripheral network nodes enhances all key aspects of edge computing, including energy efficiency, carbon footprint reduction, security, latency, privacy, offline functionality, and data management costs [5]. However, deploying intelligence at the end nodes requires careful consideration of the IoT devices inherent limitations, such as memory and computational resources impacting time performances, and energy constraints. For Microcontroller Units (MCUs), widely used in IoT, this is particularly true. Many IoT applications, such as autonomous driving [6], demand low-latency responses to be effectively reactive. Moreover, several IoT devices often operate under very limited power sources. Promising energy-efficient strategies aim to minimize consumption. For instance, index modulation [7,8] is a transmission technique that conveys additional information through the indices of available resources such as antennas, subcarriers, or time slots, and it can significantly reduce energy usage while maintaining data throughput. Nevertheless, even with advanced optimization strategies, the repetitive and frequent processing required by many applications can rapidly deplete power resources, thereby limiting device lifetime.

In recent years, Machine Learning (ML) methods [9], particularly Artificial Neural Networks (ANNs), have been increasingly deployed on IoT devices to enhance localized data processing capabilities and reduce

* Corresponding author.
E-mail addresses: andapicella@unisa.it (A. Apicella), pasquale.arpaia@unina.it (P. Arpaia), luigi.capobianco@st.com (L. Capobianco), francesco.caputo3@unina.it (F. Caputo), antonella.cioffi@st.com (A. Cioffi), antonio.esposito9@unina.it (A. Esposito), francesco.isgro@unina.it (F. Isgrò), rosanna.manzo@unina.it (R. Manzo), nicola.moccaldi@unina.it (N. Moccaldi), danilo.pau@st.com (D. Pau), ettore.toscano@st.com (E. Toscano).

https://doi.org/10.1016/j.csi.2025.104120
Received 10 January 2025; Received in revised form 2 September 2025; Accepted 21 December 2025
Available online 22 December 2025

dependency on cloud infrastructures [10,11]. It is common to refer to these devices as *tiny devices* [12] and embedded ML as *tiny machine learning* or *tiny ML* [5].

Consequently, assessing the inference time provided by the IoT hardware for a specific ANN model is crucial to ensure that the embedded system can satisfy real-time processing requirements. In this context, inference refers to the process of an ANN generating outputs based on its trained model parameters and given inputs.

Therefore, tailored energy consumption metrics are essential to ensure the alignment between the ANN implementation and the energy constraints of the targeted IoT application. To this aim, *Neural MCUs* are new edge devices embedding ANN accelerators, specifically designed to manage the trade-off between reliability, latency, cost, and power consumption [13]. Therefore, adopting standardized metrics and procedures is essential for assessing the actual performance gains achieved by neural MCUs in the context of embedded AI. Despite several frameworks and tools have been proposed to facilitate the benchmarking of tinyML models [14–16], no standardized metrics and procedures are currently defined.

Among the proposed benchmarking protocols, *MLPerf Tiny Benchmark* (MLPTB) [17] is developed by the MLCommons Association, the largest and most authoritative community aimed at improving the industrialization standardization process of machine learning [18]. MLPTB provides protocols and AI components, namely datasets and pre-trained ML models. These can act as metrological references when implemented on different hardware to assess their performance such as the inference time and the power consumption under real-world conditions. However, the MLPTB protocols exhibit some metrological weakness: (i) both the assessment of time performance and energy consumption is realized without measurement uncertainty computation, (ii) the energy consumption analysis is performed based on an approximate estimate of the average inference duration, and (iii) the impact on consumption caused by inferences is not isolated with respect to other processes.

In this paper, a new method is proposed and validated to improve MLPTB protocols to measure power consumption in MCUs running ANNs, in a rigorous metrological framework. Specifically, in Section 2 the MLPTB framework is reported, then the proposed method is presented in Section 3. Experiments and results are reported in Section 4 and discussed in Section 5.

2. Background

Several frameworks and tools have been introduced to support the benchmarking of tinyML models [14–16]. Among the available benchmarking protocols, the *MLPerf Tiny Benchmark* (MLPTB) [17], developed by the MLCommons Association [18], emerges as a key initiative.

MLPTB proposes two modalities of assessment: (i) *Performance* and (ii) *Energy*. The former measures Latency (inferences per second — IPS) and accuracy (percentage of correct predictions to all predictions ratio) through a direct USB connection between a Device Under Test (DUT) and an host computer, while the latter measures energy (micro-joules per inference). In the remainder of this section, the energy configuration mode is detailed, as it represents the central focus of this study. In the energy configuration mode (Fig. 1), an *Energy Monitor* is proposed to supply power to the DUT while measuring the current consumption. An *Input/Output Manager* is introduced to interface the *Host Computer* with the DUT and serving as an electrical-isolation proxy. Furthermore, MLPTB requires level shifters to adapt the power supply in input to the DUT (not reported in Fig. 1 to simplify the schematic as they are not essential to the discussion).

In addition to defining assessment procedures, MLPTB provides some firmware and software [19] for ML tasks on DUT. In particular, the provided firmware to be loaded onto the DUT ensures the following functionalities: (i) sending a trigger signal, (ii) enabling UART communication, (iii) generating and feeding random input data to the ANN, (iv) performing inferences, and (v) printing the prediction results. The software includes a graphical user interface that can be run on the Host Computer, allowing the initiation of the measurement and monitoring of input data. It is important to emphasize that in phase (iii) random data are generated to feed the ANN. This operation, however, does not reflect real-world applications, where the network processes sensor data in real time. Although not an intrinsic part of ANN inference, MLPTB includes this step in the performance and energy measurements. Throughout this paper, phase (iii) is explicitly distinguished from phase (iv) (i.e., inference) and is referred to as the *pre-inference* phase.

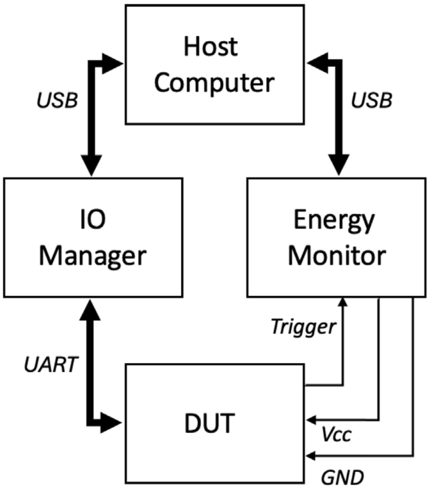


Fig. 1. Energy measurement set up proposed by MLPerf Tiny Benchmark [17, 19]. The DUT is powered by the Energy Monitor. The IO manager serves as an electrical-isolation proxy.

The energy per inference (E_{inf}) is calculated using latency information determined in the Performance phase. Specifically, the IPS is determined by taking the median value across five experiments. In each experiment, input data is provided for a duration of at least 10 s, and the number of inferences is recorded via a direct connection between the Host Computer and the DUT. Given the IPS, E_{inf} is computed as:

$$E_{inf} = \frac{I_m \times V_n}{\tau \times IPS} \tag{1}$$

where V_n is the nominal voltage, I_m is the current averaged over the fixed period τ.

3. Proposed method

The MLCommons pre-inference phase generates random numbers as input to the ANN in order to perform inference (in addition to memory operations needed to provide the input to the network). However, random number generation is hardly reproducible across different devices under test, since both the libraries and the hardware resources available on the microcontrollers for random number generation vary. In contrast, the proposed work selectively excludes the pre-inference phase from the performance and energy measurements, ensuring greater reproducibility while also providing a closer adherence to the actual operation of the device in real-world scenarios. In the following of this section, the proposed method is described. In paragraph 3.1 the circuit solution for the joint measurement of time and energy consumption is described. In paragraph 3.2 the expected impact of the method on selectivity, accuracy, and uncertainty during the energy measurement is highlighted.

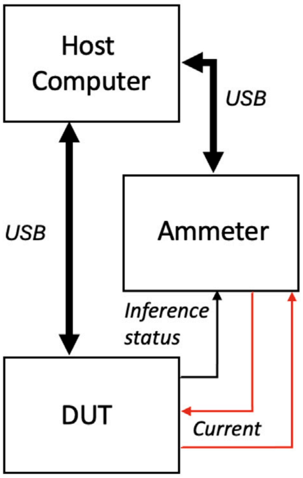


Fig. 2. Proposed energy measurement setup. The Host Computer powers the DUT and an ammeter is connected in series along the power line on the DUT (e.g. a MCU).

3.1. Circuit diagram and measurement procedure

The proposed method utilizes an ammeter that does not require powering the DUT to measure the absorbed current. The ammeter is connected in series to the microprocessor on the MCU powered by the Host Computer through the USB port (Fig. 2). This approach allows the Host Computer to perform both latency and energy measurements simultaneously. Indeed, the firmware provided by MLPTB enables the DUT to update the Host Computer on the number of completed inferences through the USB connection. Instead of computing the energy per inference as the ratio between the total energy measured in a specific time window and the number of inferences (MLPTB method), the proposed method computes the energy for each inference without considering the impact of pre-inference phase. This is obtained by modifying the firmware provided by MLPTB: the trigger is replaced by a logic signal (inference status) that goes high during an ongoing inference and returns low otherwise. The inference status signal output from the device under test is sampled by the Measurement Board (ammeter) in parallel with the current (Fig. 3.a). Two vectors of synchronously sampled data (current and inference status signal) are sent to the Host Computer. The current samples are processed, and the energy consumption is calculated only when the inference status samples indicate a low logic signal. Additionally, before and after each inference, the DUT reads the values of the Clock and Reset Management Unit (CRMU) and transmits them to the Host Computer to determine the duration of the inference. Finally, the software on the Host Computer computes the mean value of N inferences with associated uncertainty. In this work, N is set to 100. Similar to the MLPTB, the proposed firmware runs as the sole program on the MCU, with fully sequential execution and no concurrency, or interrupts. Furthermore, in the proposed method, the inference status signal is set high immediately after the pre-inference phase, and the CRMU is queried right before the inference execution. As soon as the inference completes, the CRMU is queried again, and finally the inference status is set low to signal the ammeter that the inference has finished. In Fig. 4, a flowchart describing the customized firmware behavior is reported.

3.2. Accuracy improvements

In the MLPTB, the number of inferences during the measurement time in energy mode is calculated using the IPS obtained from the previous latency measurement. This approach introduces accuracy issues because an estimator is used instead of the actual time of each inference. Furthermore, it is assumed with a non-negligible degree of approximation that the inferences are executed consecutively by the MCU, disregarding the impact of inter-inference operations that are still present. Finally, the delays in the transmission of the command for starting the measurement have a further impact on the accuracy, albeit to a very small extent. Specifically, this refers to the time taken by the CPU on the DUT to generate the trigger signal and by the Measurement Board to handle the interrupt triggered at its input pin (see Fig. 3).

In the proposed method, limiting the observation to a single inference at a time eliminates the approximation inherent in MLPTB, where the inference duration is estimated through the average of multiple successive inferences executed within a known time window. Specifically, the proposed method allows the exclusion of all energy contributions unrelated to the inference itself (e.g., data transfer operations to memory during the pre-inference phase). However, in the proposed method, the repetition of the measurement for each inference amplifies the impact of inaccuracies caused by the delay in transmitting the status signal. In contrast, the MLPTB approach mitigates this effect because the delay only occurs at the start of the measurement for multiple inferences. To address this issue, the inference duration (Δt) measurement is also performed. In the firmware for the DUT, the onboard counter is read immediately before and after the inference execution. The Δt, is used to appropriately resize the current sample vector acquired while the inference status signal is active. The current sample vector is trimmed at both ends by a number of elements (N_{trim}), calculated as follows:

$$N_{trim} = \frac{f_c}{2}\left(\frac{N_{cs}}{f_c} - \Delta t\right) \quad (2)$$

where f_c is the sampling frequency of the Ammeter, N_{cs} is the number of current samples acquired when the inference status signal is high, and Δt is the inference duration.

3.3. Uncertainty improvements

Two distinct phases should be addressed in the evaluation of uncertainty: (i) the inference time measurement, and (ii) the energy consumption assessment. In particular, an important source of uncertainty in MLPTB is due to the counting of inferences during the IPS measurement affecting inference time measurement and, consequently, also the energy consumption assessment. More deeply, the measurement window is not an integer multiple of the inference period, therefore, there is no synchronization between the end of the last inference and the end of the measurement window. This contribution can be modeled by a uniform random variable whose domain is equal to the central value inference duration Δt_m, with a standard deviation σ_{1cont} computed as:

$$\sigma_{1cont} = u_{t_1} = \frac{\Delta t_m}{2\sqrt{3}} \quad (3)$$

The uncertainty of the MLPTB method is assessed by assuming the median inference duration approximately equal to the mean. Differently, in the proposed method the counting uncertainty is determined by the fact that the inference duration is not an integer multiple of the counter period (T_c). Again, the random variable with uniform probability distribution effectively describes this aspect. The standard deviation σ_{2cont} is computed as:

$$\sigma_{2cont} = u_{t_2} = \frac{T_c}{2\sqrt{3}} \quad (4)$$

Assuming that $\Delta t_m \gg T_c$, it follows $u_{t_1} \gg u_{t_2}$ and the proposed method improves the measurement uncertainty due to counting.

Then there is the uncertainty due to the variability of the duration time of the processes between the inferences (*pre-inference* phase). The proposed method is not affected by this source of uncertainty because it excludes from the energy measurement all the processes outside the inference. Finally, both methods are exposed to the uncertainty

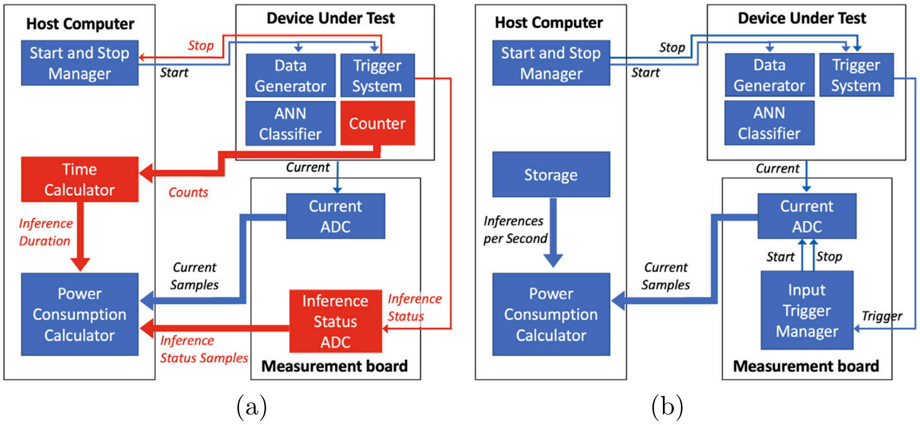


Fig. 3. Comparison between the block diagram of the proposed method (a) and ML Commons-Tiny approach (b) for energy consumption measurement. The added blocks and signals are reported in red. In the proposed method, the *Device Under Test* stops the power consumption computation after each inference. Differently, in the MLCommons-Tiny approach, the *Host Computer* stops the acquisition of current samples after a fixed time window, without distinguishing between pre-inference and inference phases. Furthermore, it computes the energy consumption (μJ per inference) based on the *Inference per Second* measured exploiting the *Performance* mode (see Section 2.) The *Counter* and the *Time Calculator* blocks are used for the measurement of the duration of each inference, while an *Inference Status ADC* minimizes the latency between the inference start and current sample consideration. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

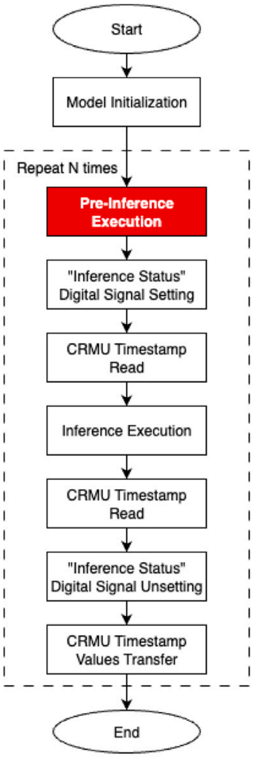


Fig. 4. Flow chart of the proposed Firmware. The pre-inference phase (in red) is excluded from both time (CRMU timestamp read) and energy assessment ("Inference Status" digital signal setting and unsetting). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of the stability of the DUT (jitter) and ammeter precision, as well as to the uncertainty of the signal transmission times between the devices involved in the measurement process. For the calculation of the measurement uncertainty, the *combined standard uncertainty* u_c is adopted, where the contribution from the type A evaluation (u_A) is integrated with the K contributions from the type B evaluations (u_{B_k}), according to the following formula [20]:

$$u_c = \sqrt{u_A^2 + u_{B_1}^2 + u_{B_2}^2 + \cdots + u_{B_K}^2}. \tag{5}$$

4. Experiments and results

In this section, a comparison between the application of the proposed and MLPTB methods is presented. In paragraph 4.1 the experimental procedure is described. The DUTs and the ammeter are presented in paragraph 4.2. Results are reported in paragraph 4.3.

4.1. Experimental procedure

The MLPTB method was implemented using two different circuit configurations for measuring inference duration and energy per inference, as described in [17]. Instead, in the proposed method the two measures were realized with the same circuital solution shown in Fig. 2. The Firmware used for MLPTB measurement was modified to allow the measurement of the single inference as described in the paragraph 3.1. The four MLPerf benchmarks were retained: (i) Anomaly Detection, (ii) Keyword Spotting, (iii) Image Classification, (iv) Visual Wake Words. Each benchmark targets a specific use case and specifies a dataset, a model, and a quality target [17].

4.2. Experimental setup

Both methods are applied on three different MCU: *STMicroelectronics STM32-H7* (Clock Frequency = 280 MHz), *STMicroelectronics STM32-U5* (Clock Frequency = 160 MHz), and *Rockchip RV1106* (Clock Frequency = 1200 MHz). The STM32H7 and the STM32U5 are general-purpose microcontrollers, the former designed for high-performance applications and the latter for ultra-low-power operation, both produced by STMicroelectronics. These devices do not have any dedicated Neural Processing Unit (NPU) hardware for ANN computation, so this part is commonly made by implemented firmware that run on main Central Process Unit (CPU). The firmware is automatically deployed using *ST EdgeAI Core Technology* and compiled through *STMCubeIDE* [21] compiler implementing all needed tools to convert, optimize, and implement ANN models on the DUT.

The evaluation boards of the STMicroelectronics *Nucleo-STM32H7* with STM32H7 microcontroller and *B-U585I-IOT02 A Discovery Kit* with STM32U5 microcontroller were chosen for the experimental setup

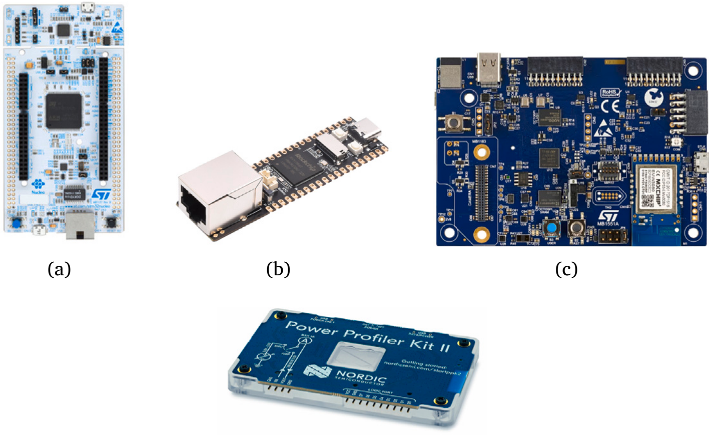

(a) (b) (c) (d)

Fig. 5. Hardware components used in the experiments: (a) H7 board with STM32H7 MCU, (b) Luckfox Pico Pro Max with Rockchip RV1106 SoC, (c) B-U585I-IOT02 A Discovery Kit with STM32U5 MCU, and (d) Power Profiler Kit II ammeter.

(Figs. 5(a), 5(c)). They include a connector in series to the MCU's power supply line allowing an ammeter to be inserted to assess the power consumption of the DUT under operating conditions.

The RV1106 is a System on Chip (SoC) produced by Rockchip Electronics. This device has a dedicated NPU hardware, so the computation of ANN models are made by hardware, and the software shall only allocate necessary data into a dedicated memory area. While STM32 microcontrollers operate without an operating system, RV1106 requires the use of an operating system given its CPU architecture. Ubuntu 22.04 RT [22] was therefore installed to minimize execution timing uncertainties.

The software is deployed using *RKNN Toolkit* compiler that implements all needed tools to convert, optimize, and implement ANN models on the device. The evaluation board with Rockchip RV1106 chosen for the experimental setup is the *Luckfox Pico Pro Max* (Fig. 5(b)). The ammeter is inserted between USB-C main supply and the SoC's power supply line in order to assess the power consumption of device under operative conditions.

The measurement board used for the power assessment is the Power Profiler Kit II (PPKII) produced by Nordic Semiconductor (Fig. 5(d)). This device is composed by an ammeter and a 8-bits digital sampler synchronized with the same time base. It can work into two different modes that affect the only ammeter component:

- *Source Meter*: With this mode, the internal ammeter is linked to a power supply generator that can be used to provide the power supply to DUT. This mode was adopted for the MLPTB implementation
- *Ammeter Mode*: With this mode, the instrument works as a pure ammeter and the power supply of DUT can be provided externally. This mode was implemented in the proposed method application.

For both modes, the device was metrologically characterized under operating conditions of 20–30 °C (the same conditions used for all experiments), exhibiting an uncertainty of less than 2%.

4.3. Results

For the proposed method, a characterization of the CRMU query latency was carried out on all devices. A modified version of the same firmware used for the energy consumption assessment was employed. Specifically, an additional CRMU query was appended directly after the preceding one, making it consecutive to the two already present. The CRMU query latency was measured as the difference between the counter values returned by two consecutive CRMU readings. On each board, 30 experiments were performed, each providing two latency values. For each board, the mean value and type A uncertainty were computed. In the worst case, namely the Rockchip, the latency was found to be 7 ± 4 CPU clock cycles (2 ± 1 for the other two boards), which corresponds to only a few nanoseconds. Tables 1, 2, and 3 present the results of inference duration (Δt) assessments conducted using both the MLPTB and the proposed methods. The results are reported for the Rockchip RV1106, STM32H7, and STM32U5, respectively, with varying ANN models. Concerning uncertainty computation, the MLPTB method does not provide strategies for calculating measurement uncertainty and, in this work, it was computed by referring to the sole contribution of the counting inferences (Eq. (2)). In the proposed method, since the Clock and Reset Management Unit (CRMU) of the MCUs is employed for inference time measurement, the type A uncertainty is combined with type B contributions arising from counting uncertainty, system clock stability (jitter), and the response time required by the CRMU to be queried and to return a value. For all the considered microcontrollers, the type B contribution was found to be dominated by the counting uncertainty, computed using formula (4), and equal to 289 ns. The jitter contribution is at least three orders of magnitude smaller at room temperature (between 20 °C and 30 °C) [23–25]. Similarly, the uncertainty related to the CRMU response time, characterized in this work for all three microcontrollers, was found to be equal to 1 CPU clock cycle. In the worst case, i.e., considering the STM32U5 device with the lowest CPU clock frequency, this contribution was on the order of nanoseconds. Therefore, the overall evaluated uncertainty corresponds to the joint contribution of type A and type B, with the latter coinciding with the counting uncertainty, according to:

$$u_t = \sqrt{u_A^2 + u_B^2} \tag{6}$$

To propagate the measurement uncertainty of the Δt on the energy per inference (E_{inf}) measurement, a constant power P is assumed during the inference time, obtaining the following propagation formula:

$$E_{inf} = P\Delta t \Rightarrow u_e = Pu_d \tag{7}$$

where u_e is the energy per inference measurement uncertainty. With respect to the energy consumption estimation, an additional uncertainty source arises from the measuring instrument, i.e., the ammeter employed. For both methods, an instrumental uncertainty of 2% was considered, after a metrological characterization performed under operational conditions at room temperature (between 20 °C and 30 °C). The

Table 1
Comparison of central value (m_t) and uncertainty[a] (u_t) of inference duration (expressed in ms) assessed by MLCommons and proposed methods on Rockchip RV1106 at varying of neural models.

Method	Visual Wake Words		Image Classification		Keyword Spotting		Anomaly Detection	
	m_t	u_t	m_t	u_t	m_t	u_t	m_t	u_t
Proposed	0.820	0.006	0.415	0.012	0.400	0.008	0.558	0.033
MLPTB	0.815	0.235	0.414	0.120	0.371	0.107	0.350	0.101

[a] In MLPTB, the counting uncertainty was taken into account.

Table 2
Comparison of central value (m_t) and uncertainty[a] (u_t) of inference duration (expressed in ms) assessed by MLCommons and proposed methods on STM32H7 microcontroller at varying of neural models.

Method	Visual Wake Words		Image Classification		Keyword Spotting		Anomaly Detection	
	m_t	u_t	m_t	u_t	m_t	u_t	m_t	u_t
Proposed	29.656	0.003	49.941	0.001	14.860	0.001	1.690	0.002
MLPTB	29.600	8.545	51.900	14.982	15.400	4.446	1.800	0.520

[a] In MLPTB, the Counting Uncertainty was taken into account.

Table 3
Comparison of central value (m_t) and uncertainty[a] (u_t) of inference duration (expressed in ms) assessed by MLCommons and proposed methods on STM32U5 microcontroller at varying of neural models.

Method	Visual Wake Words		Image Classification		Keyword Spotting		Anomaly Detection	
	m_t	u_t	m_t	u_t	m_t	u_t	m_t	u_t
Proposed	78.447	0.002	133.280	0.002	48.060	0.001	4.910	0.002
MLPTB	71.600	20.669	128.200	37.008	38.600	11.143	4.800	1.386

[a] In MLPTB, the Counting Uncertainty was taken into account.

Table 4
Comparison of central value (m_t) and uncertainty[a] (u_e) of energy (expressed in μJ) assessed by MLCommons and proposed methods on Rockchip RV1106 at varying of neural models.

Method	Visual Wake Words		Image Classification		Keyword Spotting		Anomaly Detection	
	m_t	u_e	m_t	u_e	m_t	u_e	m_t	u_e
Proposed	380	13	193	15	165	9	222	11
MLPTB	373	108	183	53	159	46	148	43

[a] In MLPTB, the counting uncertainty was propagated into the energy measurements.

Table 5
Comparison of central value (m_t) and uncertainty[a] (u_e) of energy (expressed in μJ) assessed by MLCommons and proposed methods on STM32H7 microcontroller at varying of neural models.

Method	Visual Wake Words		Image Classification		Keyword Spotting		Anomaly Detection	
	m_t	u_e	m_t	u_e	m_t	u_e	m_t	u_e
Proposed	4386	88	7536	151	2202	44	236	6
MLPTB	3699	1068	6311	1822	1870	540	221	64

[a] In MLPTB, the counting uncertainty was propagated into the energy measurements.

final uncertainty was thus obtained by applying the following formula:

$$u_e = \sqrt{u_{t_p}^2 + u_s^2} \tag{8}$$

where u_{t_p} denotes the inference time measurement uncertainty u_t propagated through the functional relation used for energy computation (see formula), and u_s represents the instrumental uncertainty of the ammeter. The measurement uncertainty obtained for the proposed method appears for all tested devices to be very low compared to the uncertainty of the MLPTB method.

In Tables 4, 5, and 6 a comparison between results of energy per inference assessment by MLPTB and proposed methods are reported for the three DUTs. On the Rockchip RV1106, the proposed method measures an inference energy value that is, on average, 15% higher than that obtained with MLPTB, while improving the uncertainty by a factor of 6. In the case of a STM32H7 inference energy assessment grows by 16% while the uncertainty improves by a factor of 12. Notably, the inference energy assessment on the STM32U5 shows contrasting trends: for two networks, the measured consumption is higher with the proposed method, while for the other two networks it is higher with MLCommons. Regarding the uncertainty, the proposed method reduces it by a factor of 12.

5. Discussion

The contrasting trends from energy assessment on STM32U5 provide an opportunity to discuss the relationship between the two methods in terms of metrological accuracy. The MLCommons method extracts a central Inference Per Second value based on five experiments, whereas our method computes a central value as the mean over 100 acquisitions. Given the large uncertainty of the MLPTB method and the limited number of experiments, the calculated central value is unlikely to be a reliable estimator of the true value of the measured quantity [26]. The comparison of mean values obtained with the two methods is limited by the large difference in their associated uncertainties. The less precise method exhibits an uncertainty up to two orders

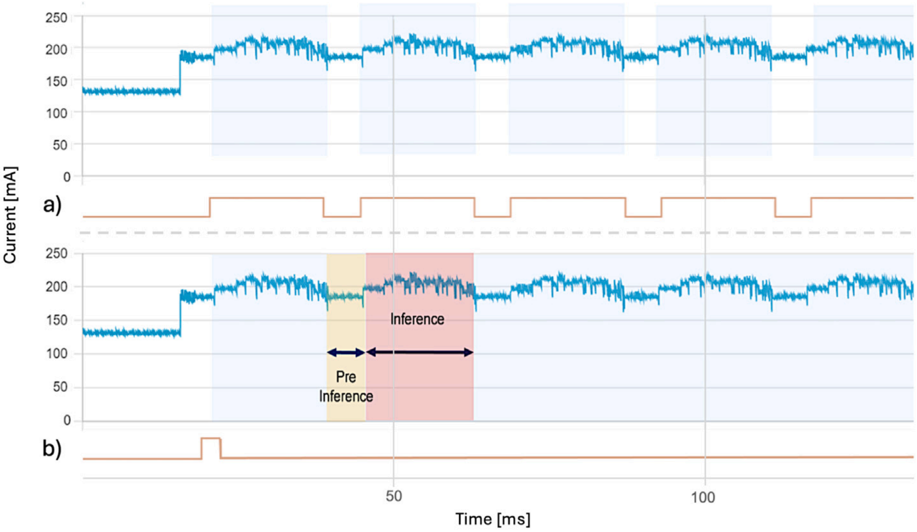


Fig. 6. Temporal diagram of current values acquired from MCU during ANN operations. Orange traces represent (a) the *inference status signal* in the proposed method and (b) the *trigger signal* in the MLPTB method. The windows used for energy consumption estimation are highlighted in light blue. Specifically, the proposed method (a) considers only the current samples acquired during each neural network inference phase, whereas the MLPTB method (b) also includes the energy contribution of pre-inference phases (light yellow window). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

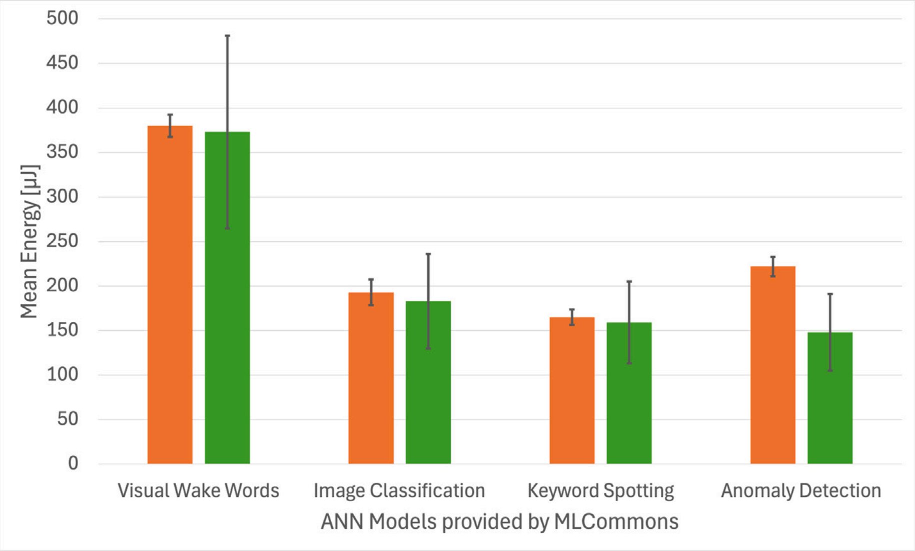


Fig. 7. Comparison between proposed method (orange) and MLPTB (green) in Energy per inference Assessment on the Rockchip RV1106, at varying th Models provided by MLCommons. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Comparison of central value (m_t) and uncertainty[a] (u_e) of energy (expressed in µJ) assessed by MLCommons and proposed methods on STM32U5 microcontroller at varying of neural models.

Method	Visual Wake Words		Image Classification		Keyword Spotting		Anomaly Detection	
	m_t	u_e	m_t	u_e	m_t	u_e	m_t	u_e
Proposed	2362	47	3249	65	1184	27	116	3
MLPTB	1921	556	3384	980	1004	291	121	35

[a] In MLPTB, the counting uncertainty was propagated into the energy measurements.

of magnitude higher than the other, rendering direct statistical comparisons of the means largely insignificant. Observed differences may therefore primarily reflect the inherent variability of the less accurate method rather than genuine differences in the measured phenomenon. However, it is important to note that the proposed method provides greater selectivity by excluding the pre-inference phase (characterized by low energy consumption) from the calculation (Fig. 6). This prevents underestimation of the actual energy consumption, which may occur when using the MLPTB method.

Finally the Figs. 7, 8, and 9 present the histograms of Energy per Inference assessment with the two methods on Rockchip RV1106, STM32H7, and STM32U5, respectively. The orange bars (proposed

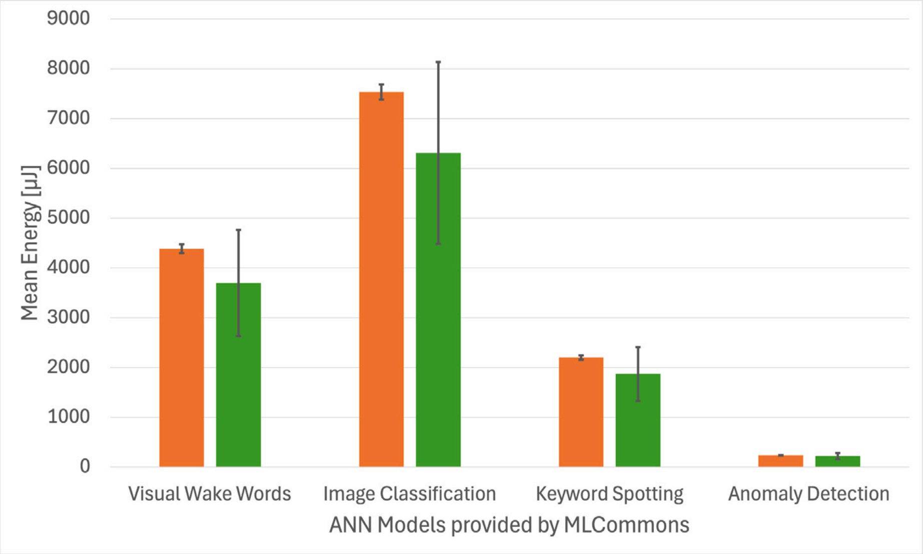


Fig. 8. Comparison between proposed method (orange) and MLPTB (green) in Energy per inference Assessment on the STM32 H7, at varying th Models provided by MLCommons. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

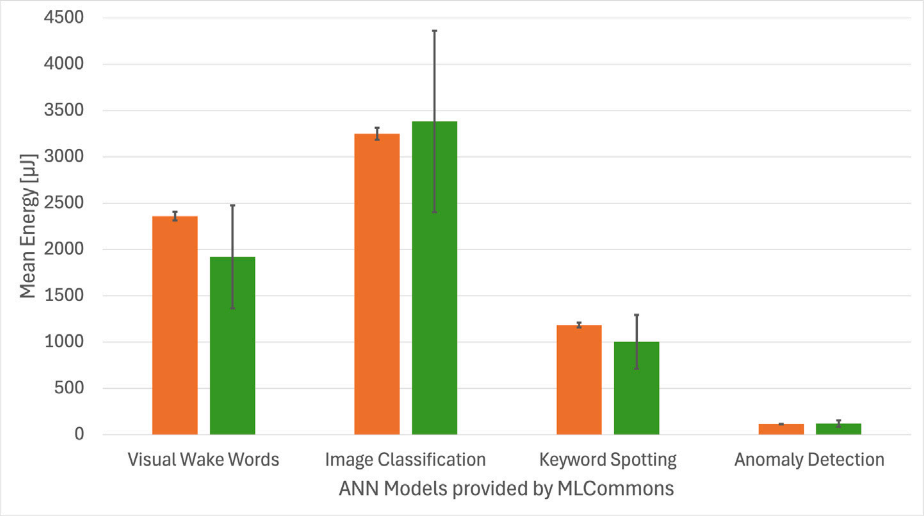


Fig. 9. Comparison between proposed method (orange) and MLPTB (green) in Energy per inference Assessment on the STM32 U5, at varying th Models provided by MLCommons. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

method) are generally higher than the green bars (MLPTB). However, comparing the mean values measured by the two methods is challenging due to the large uncertainty intervals (error bars) associated with MLPTB. Nevertheless, the differences in error bar lengths confirm the improved precision of the proposed method.

The metrological improvements introduced in this work have direct consequences for the practical adoption of embedded AI. First, more accurate and reproducible energy assessments enhance the reliability of benchmarking, enabling fair comparisons among devices and supporting informed selection of hardware for battery-powered applications, where autonomy is a critical design constraint. Second, the improved accuracy in energy characterization facilitates more precise sizing of power supply components, which is essential for ensuring efficiency, stability, and cost-effectiveness in embedded deployments. Finally, the refined timing characterization allows designers to better estimate inference latency, a key parameter for real-time and safety-critical applications.

6. Conclusions

A new method for assessing power consumption of edge devices such as MCUs running ANNs is presented, claiming metrological improvements over the MLPerf Tiny Benchmark. Unlike MLPTB, the proposed method calculates the duration and energy consumption of each individual inference performed by the Device Under Test. Through an appropriate circuit and firmware design, the method measures only the energy consumed by the inference, excluding other operations from the computation. This approach not only enhances the selectivity and accuracy of the measurement process but also reduces measurement uncertainty. Instead of counting the number of inferences over a fixed interval, as MLPTB does, the proposed method counts the number of ticks from the counter of the DUT during a single inference execution. On a NPU powered microcontroller, the proposed method improves measurement uncertainty by a factor of 6. In the case of two general-purpose microcontrollers (high-performance and ultra-low-power), the measurement uncertainty improves by a factor of 12.

CRediT authorship contribution statement

Andrea Apicella: Writing – review & editing, Methodology, Conceptualization. **Pasquale Arpaia:** Writing – review & editing, Methodology, Conceptualization. **Luigi Capobianco:** Writing – review & editing, Methodology, Conceptualization. **Francesco Caputo:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Antonella Cioffi:** Writing – review & editing, Methodology, Conceptualization. **Antonio Esposito:** Writing – review & editing, Methodology, Conceptualization. **Francesco Isgrò:** Writing – review & editing, Methodology, Conceptualization. **Rosanna Manzo:** Writing – review & editing, Methodology, Conceptualization. **Nicola Moccaldi:** Writing – review & editing, Methodology, Conceptualization. **Danilo Pau:** Writing – review & editing, Methodology, Conceptualization. **Ettore Toscano:** Writing – review & editing, Methodology, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was carried out within the DHEAL-COM project (ID: PNC-E3-2022-23683267 PNC – HLS – DH; CUP: E63C22003790001), which was financially supported by the Italian Ministry of Health through the Complementary National Plan (CNP) to the PNRR. This publication reflects only the authors' view and the Italian Ministry of Health is not responsible for any use that may be made of the information it contains.

Data availability

Data will be made available on request.

References

[1] R. Chataut, A. Phoummalayvane, R. Akl, Unleashing the power of IoT: A comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and industry 4.0, Sensors 23 (16) (2023) 7194.

[2] Q. Ma, H. Tan, T. Zhou, Mutual authentication scheme for smart devices in IoT-enabled smart home systems, Comput. Stand. Interfaces 86 (2023) 103743.

[3] C.-W. Shih, C.-H. Wang, Integrating wireless sensor networks with statistical quality control to develop a cold chain system in food industries, Comput. Stand. Interfaces 45 (2016) 62–78.

[4] S.B. Baker, W. Xiang, I. Atkinson, Internet of things for smart healthcare: Technologies, challenges, and opportunities, IEEE Access 5 (2017) 26521–26544.

[5] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, A.S. Hafid, A comprehensive survey on tinyml, IEEE Access (2023).

[6] M. Cunneen, M. Mullins, F. Murphy, Autonomous vehicles and embedded artificial intelligence: The challenges of framing machine driving decisions, Appl. Artif. Intell. 33 (8) (2019) 706–731.

[7] J. Li, S. Dang, M. Wen, Q. Li, Y. Chen, Y. Huang, W. Shang, Index modulation multiple access for 6G communications: Principles, applications, and challenges, IEEE Netw. 37 (1) (2023) 52–60.

[8] M. Wen, B. Zheng, K.J. Kim, M. Di Renzo, T.A. Tsiftsis, K.-C. Chen, N. Al-Dhahir, A survey on spatial modulation in emerging wireless systems: Research progresses and applications, IEEE J. Sel. Areas Commun. 37 (9) (2019) 1949–1972.

[9] M.I. Jordan, T.M. Mitchell, Machine learning: Trends, perspectives, and prospects, Science 349 (6245) (2015) 255–260.

[10] S. Mishra, J. Manda, Improving real-time analytics through the internet of things and data processing at the network edge, J. AI Assist. Sci. Discov. 4 (1) (2024) 184–206.

[11] M. De Donno, K. Tange, N. Dragoni, Foundations and evolution of modern computing paradigms: Cloud, IoT, edge, and fog, IEEE Access 7 (2019) 150936–150948.

[12] D.P. Pau, P.K. Ambrose, F.M. Aymone, A quantitative review of automated neural search and on-device learning for tiny devices, Chips 2 (2) (2023) 130–141.

[13] C.-T. Lin, P.X. Huang, J. Oh, D. Wang, M. Seok, iMCU: A 102-μJ, 61-ms digital in-memory computing-based microcontroller unit for edge TinyML, in: 2023 IEEE Custom Integrated Circuits Conference, CICC, IEEE, 2023, pp. 1–2.

[14] S. Gal-On, M. Levy, Exploring coremark a benchmark maximizing simplicity and efficacy, Embed. Microprocess. Benchmark Consortium (2012).

[15] P. Torelli, M. Bangale, Measuring Inference Performance of Machine-Learning Frameworks on Edge-Class Devices with the Mlmark Benchmark, Techincal Report, 2021, Available Online: https://www.eembc.org/techlit/articles/MLMARK-WHITEPAPERFINAL-1.pdf. (Accessed on 5 April 2021).

[16] B. Sudharsan, S. Salerno, D.-D. Nguyen, M. Yahya, A. Wahid, P. Yadav, J.G. Breslin, M.I. Ali, Tinyml benchmark: Executing fully connected neural networks on commodity microcontrollers, in: 2021 IEEE 7th World Forum on Internet of Things, WF-IoT, IEEE, 2021, pp. 883–884.

[17] C. Banbury, V.J. Reddi, P. Torelli, J. Holleman, N. Jeffries, C. Kiraly, P. Montino, D. Kanter, S. Ahmed, D. Pau, et al., Mlperf tiny benchmark, 2021, arXiv preprint arXiv:2106.07597.

[18] MLCommons, 2024, URL: https://mlcommons.org/benchmarks/inference-tiny/.

[19] Performance mode vs. Energy mode, 2022, URL: https://github.com/eembc/energyrunner?tab=readme-ov-file#performance-mode-vs-energy-mode.

[20] B.N. Taylor, C.E. Kuyatt, Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results, NIST Technical Note 1297, National Institute of Standards and Technology (NIST), Gaithersburg, MD, 2020, http://dx.doi.org/10.6028/NIST.TN.1297-2020.

[21] STMCubeIDE, 2022, URL: https://stm32ai.st.com/stm32-cube-ai/.

[22] Ubuntu 12 RT, 2012, Real-time variant of Ubuntu 12, Canonical Ltd. https://ubuntu.com/real-time. Canonical Ltd.

[23] STMicroelectronics, STM32H753xI - 32-bit Arm® Cortex®-M7 480MHz MCUs, 2MB flash, 1MB RAM, 46 com. and Analog Interfaces, Crypto - Datasheet - Production Data, Datasheet DS12117 Rev 9, STMicroelectronics, 2023, p. 358, URL: https://www.st.com/resource/en/datasheet/stm32h753vi.pdf. (Accessed 21 August 2025).

[24] STMicroelectronics, STM32U575xx - Ultra-low-power Arm® Cortex®-M33 32-bit MCU+TrustZone®+FPU, 240 DMIPS, up to 2 MB Flash memory, 786 KB SRAM - Datasheet - production data, Datasheet DS13737 Rev 10, STMicroelectronics, 2024, p. 346, URL: https://www.st.com/resource/en/datasheet/stm32u575ag.pdf. (Accessed 21 August 2025).

[25] UEC Electronics, AR4236–AR4237 Luckfox Pico Pro/Max Datasheet, Datasheet, UEC Electronics, 2024, URL: https://uelectronics.com/wp-content/uploads/2024/07/AR4236-AR4237-Luckfox-Pico-Pro-Max-Datasheet.pdf. (Accessed 21 August 2025).

[26] I. BIPM, I. IFCC, I. ISO, O. IUPAP, Evaluation of measurement data—guide to the expression of uncertainty in measurement, JCGM 100: 2008 GUM 1995 with minor corrections, Jt. Comm. Guides Metrol. 98 (2008).